This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

1. (Previously Presented) A method of simultaneously optimizing code for at least two target machines, comprising:

abstracting a rule of instruction scheduling for each of said at least two target machines;

generating a hypothetical machine model of a hypothetical machine based on said rule of instruction scheduling for each of said at least two target machines, wherein a rule of instruction scheduling for said hypothetical machine is a restrictive set of said rule of instruction scheduling for each of said at least two target machines; and

targeting said hypothetical machine, wherein said hypothetical machine model is capable of operating on each of said at least two target machines.

2. (Canceled)

3. (Previously Presented) The method of claim 1 further including:

detecting a conflict between said rule of instruction scheduling for each of said at least two target machines; and

resolving said conflict.

4. (Previously Presented) The method of claim 3 wherein resolving said conflict includes selecting a less damaging option for said detected conflict.

5. (Previously Presented) The method of claim 3 wherein said detected conflict corresponds to an inherent conflict between said rule of instruction scheduling for each of said at least two target machines.

6. (Previously Presented) The method of claim 1 further including:

modeling each of said at least two target machines; and

retrieving scheduling information corresponding to each of said at least two target machines.

7. (Original) The method of claim 1 wherein said at least two target machines include an UltraSPARC-II configured to operate at a speed of 360 MHz and an UltraSPARC-III configured to operate at a speed of 600 MHz.

8. (Previously Presented) A method of simultaneously optimizing code for at least two target machines, comprising:

retrieving scheduling information corresponding to each of said at least two target machines;

abstracting a rule of instruction scheduling for each of said at least two target machines;

generating a hypothetical machine model of a hypothetical machine based on said rule of instruction scheduling for each of said at least two target machines, wherein a rule of instruction scheduling for said hypothetical machine is a restrictive set of said rule of instruction scheduling for each of said at least two target machines; and

targeting said hypothetical machine, wherein said hypothetical machine model is capable of operating on each of said at least two target machines.

9. (Previously Presented) The method of claim 8 further including:

detecting a conflict between said rule of instruction scheduling for each of said at least two target machines; and

resolving said conflict.

10. (Canceled)

11. (Previously Presented) The method of claim 9 wherein resolving said conflict includes selecting a less damaging option for said detected conflict.

12. (Previously Presented) The method of claim 9 wherein said detected conflict corresponds to an inherent conflict between said rule of instruction for each of said at least two target machines.

13. (Previously Presented) An apparatus for simultaneously optimizing code for at least two target machines, comprising:

means for abstracting a rule of instruction scheduling for each of said at least two target machines;

means for generating a hypothetical machine model of a hypothetical machine based on said rule of instruction scheduling for each of said at least two target machines, wherein a rule of instruction scheduling for said hypothetical machine is a restrictive set of said rule of instruction scheduling for each of said at least two target machines; and

means for targeting said hypothetical machine, wherein said hypothetical machine model is capable of operating on each of said at least two target machines.

14. (Canceled)

15. (Previously Presented) The apparatus of claim 13 further including:

means for detecting a conflict between said rule of instruction scheduling for each of said at least two target machines; and

means for resolving said conflict.

16. (Previously Presented) The apparatus of claim 15 wherein said resolving means includes means for selecting a less damaging option for said detected conflict.

17. (Previously Presented) The apparatus of claim 15 wherein said detected conflict corresponds to an inherent conflict between said rule of instruction scheduling for each of said at least two target machines.

18. (Original) The apparatus of claim 13 further including:

means for modeling each of said at least two target machines; and

means for retrieving scheduling information corresponding to each of said at least two target machines.

19. (Previously Presented) An apparatus for simultaneously optimizing code for at least two target machines, comprising:

means for receiving scheduling information corresponding to each of said at least two target machines;

means for abstracting a rule of instruction scheduling for each of said at least two target machines;

means for generating a hypothetical machine model of a hypothetical machine based on said rule of instruction scheduling for each of said at least two target machines, wherein a rule of instruction scheduling for said hypothetical machine is a restrictive set of said rule of instruction scheduling for each of said at least two target machines; and

means for targeting said hypothetical machine, wherein said hypothetical machine model is capable of operating on each of said at least two target machines.

20. (Previously Presented) An apparatus for simultaneously optimizing code for a plurality of target machines, comprising:

means for modeling a plurality of target machines;

means for retrieving scheduling information corresponding to each of said plurality of target machines;

means for abstracting a rule of instruction scheduling for each of said plurality of target machines;

means for generating a hypothetical machine model of a hypothetical machine based on said rule of instruction scheduling for each of said at least two target machines, wherein a rule of instruction scheduling for said hypothetical machine is a restrictive set of said rule of instruction scheduling for each of said at least two target machines; and

means for targeting said hypothetical machine, wherein said hypothetical machine model is capable of operating on each of said at least two target machines.

means for detecting a conflict between said rule of instruction scheduling for each of said plurality of target machines; and

means for resolving said conflict.